WHAT IS CLAIMED IS

1 A method for converting a computer program into an executable object having symbol references that can be redirected at run-time, the method executing on a computer system, the computer system including a processor and storage device, the computer system further including a computer program having symbolic references to original definitions having original names, the method comprising the steps of:

identifying one or more of the original names;

renaming one or more original names used in the computer program with new names; and

creating an association between the original names and the new names so that symbolic references to the original names invoke a reference to the new names.

2. The method of claim 1, further comprising the step of

storing information about the associations in a table format.

3. The method of claim 2, further comprising the steps of

causing a symbolic reference to reference an entry in the table; and

associating a pointer to an original definition with the entry.

4. The method of claim 1, wherein a compiler is used to compile the computer program, the method further comprising the step of

using information generated by the compiler to perform the step of identifying original names in the computer program.

5. The method of claim 1, wherein a symbolic reference is to a data structure.

6. The method of claim 1, wherein a symbolic reference is to a program instruction.

7. The method of claim 1, wherein a symbolic reference is to a resource.

8. The method of claim 1, wherein a symbolic reference is to an object.

9. The method of claim 1, wherein the Microsoft Developer's Environment is used to compile the computer program, wherein the Microsoft Developer's Environment includes utilites for generating information about symbolic references in the computer program, the method further comprising the steps of

using information from one or more files to generate one or more auxiliary files that include information on original names; and

using the auxiliary file to compile an add-on module for execution in conjunction with the computer program.

10. The method of claim 1, wherein the Microsoft Developer's Environment provides for compiling the computer program by using a linker that generates a .map file, the method further comprising

wherein the step of using information from one or more files to generate an auxiliary file includes the substep of using the .map file to derive a .def file that is included as at least a portion of the auxiliary file.

11. A method for providing run-time modification of functionality in a computer program that has a substitute reference for one or more symbolic references used in the computer program, the method executing on a computer system, the computer system including a processor and storage device, the method including the steps of:

loading the computer program into the computer system;

loading a module that includes an item definiton into the computer system; and

executing a process to associate the substitute reference with the item definition.

12. The method of claim 11, wherein the method further comprises the steps of:

associating the substitute reference with the item definition by executing script language instructions.

13. The method of claim 11, wherein the method further comprises the steps of:

associating the substitute reference with the item definition during execution of the computer program by concurrently executing script language *instructions* to control the associations.